Частное учреждение образования

«Колледж бизнеса и права»

Лабораторная работа №4

Учащийся Е.А. Чужавко.

группа: Т-992

2021

Текст задачи 2

Создать 3 класса(базовый) и 2 предка которые описывают некоторых работников с почасовой оплатой (один из предков) и фиксированной оплатой (второй предок). Описать в базовом классе абстрактный метод для расчета среднемесячной зарплаты. Для «почасовиков» формула для расчета такая: «среднемесячная зарплата = 20.8\*8\*ставка в час», для работников с фиксированной оплатой «среднемесячная зарплата = фиксированной месячной оплате».

// Main

package core;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

//import java.io.\*;

public class Main {

public static void main(String[] args) {

Admin employ1 = new Admin("Matvey", 50.48);

Developer employ2 = new Developer("Artem", 1000.12);

employ1.display();

employ2.display();

employ1.srednInMonth();

employ2.srednInMonth();

Admin[] array = new Admin[8];

// Заполнено по возрастанию

for (int i = 0;i < array.length;i++){

array[i] = new Admin("name"+i, i);

}

// Упорядочиваю по зарплате

for (int i = 0; i < array.length; i++) {

for (int j = 0; j < array.length-1; j++) {

if (array[j].money < array[j + 1].money) {

Admin b = new Admin(array[j].name, array[j].money);

array[j] = array[j + 1];

array[j + 1] = new Admin(b.name, b.money);

}

}

}

System.out.println("Первые 5: ");

try(FileWriter writer = new FileWriter("test\_4.txt", false))

{

// Печатаю для проверки

for (int i = 0; i < 5; i++) {

String s = array[i].name + "\_\_" + array[i].money + " index = " + i;

writer.write(s);

writer.append('\n');

writer.flush();

System.out.println(s);

}

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

System.out.println("Последние 3: ");

for (int i = array.length-1; i > array.length-3-1; i--) {

System.out.println(array[i].name + "\_\_" + array[i].money + " index = " + i);

}

}

}

Результат работы программы показан на рисунке 1.

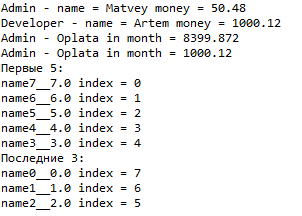


Рисунок 1

// admin

package core;

public class Admin extends Employee{

// А он принимает почасовую оплату

public Admin(String name, double money) {

super(name, money);

}

@Override

public void display() {

System.out.println("Admin - name = " + super.name + " money = " + super.money);

}

@Override

public void srednInMonth() {

System.out.println("Admin - Oplata in month = " + 20.8\*8\*super.money);

}

}

// developer

package core;

public class Developer extends Employee{

// У него будет фиксированная оплата

public Developer(String name, double money) {

super(name, money);

}

@Override

public void display() {

System.out.println("Developer - name = " + super.name + " money = " + super.money);

}

@Override

public void srednInMonth() {

System.out.println("Admin - Oplata in month = " + super.money);

}

}

// Employee

package core;

public abstract class Employee {

String name;

double money;

Employee(String name, double money){

this.name = name;

this.money = money;

}

public String getName() { return name; }

public abstract void display();

public abstract void srednInMonth();

}

Текст задачи 2

Создайте собственного питомца, который является наследником приведенных в приложении абстрактного класса и интерфейса. (6. Попугай)

// Main

package com.example;

import com.example.domain.\*;

import com.example.domain.Animal;

import com.example.domain.Cat;

public class EmployeeTest {

private static void printEmployee(Employee emp) {

System.out.println("Employee ID: " + emp.getEmpId());

System.out.println("Employee Name: " + emp.getName());

System.out.println("Employee Soc Sec # " +

emp.getSsn());

System.out.println("Employee salary: " +

emp.getSalary());

}

public static void main(String[] args) {

Engineer eng = new Engineer(101, "Jane Smith", "012-34-5678",120\_345.27);

Manager mgr = new Manager(207, "Barbara Johnson", "054-12-2367", 109\_501.36, "US Marketing");

Admin adm = new Admin(304, "Bill Munroe", "108-23-2367",75\_002.34);

Director dir = new Director(12, "Susan Wheeler", "099-45-2340", 120\_567.36, "Global Marketing", 1\_000\_000.00);

printEmployee(eng);

printEmployee(mgr);

printEmployee(adm);

printEmployee(dir);

}

}

// POPUGAI

package com.example.domain;

public class POPUGAI extends Animal implements Pet {

float speedFly = (float) 12.3;

public POPUGAI(String name) {

super(name);

// TODO Auto-generated constructor stub

}

@Override

public String getName() {

// TODO Auto-generated method stub

return null;

}

@Override

public void setName(String name) {

// TODO Auto-generated method stub

}

@Override

public void play() {

System.out.println("Я летаю! " + speedFly);

}

@Override

public

void eat() {

System.out.println("Я кушаю корм для попугаев!");

}

}

// pet

package com.example.domain;

interface Pet {

public String getName();

public void setName (String name);

public void play ();

}

// manager

package com.example.domain;

public class Manager extends Employee {

private String deptName;

public Manager(int empId, String name, String ssn, double salary, String deptName) {

super(empId, name, ssn, salary);

this.deptName = deptName;

}

public String getDeptName() {

return deptName;

}

}

// fish

package com.example.domain;

public class Fish extends Animal implements Pet {

Fish(String name) {

super(name);

// TODO Auto-generated constructor stub

}

@Override

public String getName() {

return getName();

}

@Override

public void setName(String name) {

this.name = name;

} @Override

public void play() {

System.out.println("Рыбка просто плавает");

}

@Override

public

void eat() {

// TODO Auto-generated method stub

} }

// engineer

package com.example.domain;

public class Engineer extends Employee {

public Engineer(int empId, String name, String ssn, double salary) {

super(empId, name, ssn, salary);

}

}

// employee

package com.example.domain;

public class Employee {

private int empId;

private String name;

private String ssn;

private double salary;

public Employee(int empId, String name, String ssn, double salary) {

this.empId = empId;

this.name = name;

this.ssn = ssn;

this.salary = salary;

}

// get

public int getEmpId()

{

return empId;

}

public String getName()

{

return name;

}

public String getSsn()

{

return ssn;

}

public double getSalary()

{

return salary;

}

// set

public void setName(String x)

{

name = x;

}

// Метод увеличения зарплаты

public void raiseSalary(double increase){

if (increase>0){

salary += increase;

}

}

}

// director

package com.example.domain;

public class Director extends Manager {

private double budget;

public Director(int empId, String name, String ssn, double

salary, String deptName, double budget) {

super(empId, name, ssn, salary, deptName);

this.budget = budget;

}

public double getBudget() {

return budget;

}

}

// cat package com.example.domain;

public class Cat extends Animal implements Pet {

Cat(String name) {

super(name);

// TODO Auto-generated constructor stub

}

private String name;

@Override

public String getName() {

return getName();

}

@Override

public void setName(String name) {

this.name = name;

}

@Override

public void play() {

System.out.println(name + " любит играть с веревочкой");

}

@Override

public

void eat() {

// TODO Auto-generated method stub

} }

// animal package com.example.domain;

public abstract class Animal {

String name;

Animal(String name){

this.name = name;

}

public abstract void eat();

}

// admin

package com.example.domain;

public class Admin extends Employee {

public Admin(int empId, String name, String ssn, double salary) {

super(empId, name, ssn, salary);

}

}

Результат работы программы показан на рисунке 2.

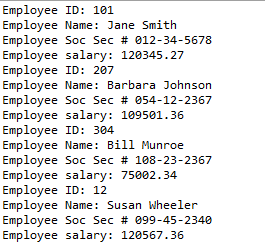


Рисунок 2